

## Examen Parcial de Programación II – Ejercicio Práctico

27 de Mayo de 2019

**Realización:** El ejercicio se realizará en la hoja de respuestas, que será lo único que se entregará. En ella se harán constar los apellidos y el nombre. Se pueden utilizar hojas de sucio aparte.

**Puntuación:** Este ejercicio tiene un peso de un 70% en la nota total del examen.

**Duración:** 1 hora y 30 minutos.

**Calificaciones:** Las notas se publicarán el día 31 de Mayo.

### ENUNCIADO

Se dispone de una clase ya realizada llamada *DatosEconomicos* que permite almacenar los datos macroeconómicos de un país en un año determinado. Su especificación es:

```
DatosEconomicos
- codigoPais : int        // código numérico que identifica el país
- nombrePais : String     // nombre del país
- pib : int               // producto interior bruto, en millones de euros
- ipc : double            // índice de precios al consumo, en porcentaje
- año : int               // año correspondiente a los datos almacenados
+ DatosEconomicos (codigoPais : int; nombrePais : String; pib : int;
                    ipc : double; año : int)
+ toString () : String
+ codigoPais () : int
+ nombrePais () : String
+ pib () : int
+ ipc () : double
+ año () : int
```

Se quiere implementar en Java la clase *HistoriaEconomica* que sirve para guardar los datos históricos de varios países a lo largo de varios años y que dispone de métodos para obtener información a partir de esta historia.

La clase tiene una lista de objetos de la clase *DatosEconomicos*, almacenados en un atributo de tipo *ArrayList<DatosEconomicos>* llamado *lista*.

Suponiendo ya implementada la excepción *FueraDeTiempo*, y que se tiene un tester llamado *HistoriaEconomicaTest* que hace pruebas sobre *HistoriaEconomica*, se pide:

#### Problema 1: (2 puntos)

Implementar en Java el siguiente método de la clase *HistoriaEconomica*:

añade (de : DatosEconomicos) throws FueraDeTiempo

PRE: El año de <de> es ~~menor~~ <sup>mayor o igual</sup> que el año del elemento más recientemente añadido a la lista.

EFFECTO: Añade <de> a la lista poniéndolo al final.

Si no se cumple la PRE lanza la excepción <FueraDeTiempo>.

**Problema 2: (2 puntos)**

Se necesita representar en el tester los siguientes datos económicos:

Código	Nombre	PIB	IPC	Año
30	GRECIA	184714 M€	0,9	2018
34	ESPAÑA	1208248 M€	1,5	2018
39	ITALIA	1753949 M€	1,1	2018

Se pide escribir el código Java para hacer lo siguiente dentro del método *main* del tester:

- Declarar y crear tres objetos de nombre *greceia*, *españa* e *italia*.
- Declarar y crear un objeto de nombre *gei* del tipo *HistoriaEconomica* y cargarlo con los datos económicos de Grecia, España e Italia.

Implementar en Java los siguientes métodos de la clase *HistoriaEconomica*:

**Problema 3: (2 puntos)**

*numeroDatos* (codigo : int) : int

POST: Retorna el número de objetos con datos económicos almacenados para el país con este <codigo>.

**Problema 4: (2 puntos)**

*hayIPCSuperior*(porcentaje : double) : boolean

POST: Determina si hay en la lista algún ipc superior a <porcentaje>.

**Problema 5: (2 puntos)**

*paisPIBMaximo* (año1, año2 : int) : int

POST: Resultado es el código del país para el que se da el PIB más grande entre los años <año1> y <año2>.

Si no hay ningún país entre los años <año1> y <año2> o si <año1> es mayor que <año2> el resultado es -1.